

# ***Modelagem e Simulação***

## ***ARENA***

**Graça Bressan**  
**LARC-PCS/EPUSP**

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Tipos de Modelagem***

- ◆ **Modelagem por Evento**
  - O comportamento do sistema é regido pela ocorrência de eventos que causam transição de estados.
- ◆ **Modelagem por Processo**
  - Os processos definem como as entidades fluem através de sistema.
  - Podem existir diferentes processos no sistema.
- ◆ **Modelagem por atividade.**
  - Ocorre um detalhamento das atividades realizadas pelas entidades.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## *Elementos da Modelagem*

### ◆ Entidades

- São criadas ou entram no sistema e percorrem o sistema até a sua saída ou destruição.

### ◆ Atributos

- Informações armazenadas junto com as entidades.

### ◆ Recursos

- São requisitados pelas entidades para realizar atividades.

### ◆ Filas

- Quando um recurso não está disponível a entidade espera em uma fila.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## *Elementos da Modelagem*

Tipo de sistema	Entidades	Atributos	Recursos	Filas
<b>Manufatura</b>	Partes	Código de peças, datas de entrega	Máquinas, trabalhadores	Filas, estoques
<b>Comunicações</b>	Mensagens	Destino, comprimento da mensagem	Nós, enlaces	Buffers
<b>Aeroporto</b>	Aviões	Número do voo Capacidade	Pistas, terminais	Filas
<b>Supermercados</b>	Compradores	Tamanho da compra	Caixas	Filas

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Arena***

- ◆ Lançado pela empresa Systems Modeling em 1993
- ◆ Sucessor do SIMAN, desenvolvido em 1982.
- ◆ Possui uma interface gráfica GUI que permite a modelagem do sistema através de módulos.
- ◆ Possui interface para Microsoft VBA permitindo integração com programas que suportam Active X.
- ◆ A versão Arena 3 Academic, disponível para uso livre de pagamento, e que será utilizada nos exemplos, possui limitações no número de entidades que podem ser criadas.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Ferramentas do Arena***

- ◆ **Arena**: A ferramenta de modelagem e simulação
- ◆ **Input Analyser**: Realiza a análise estatística dos dados de entrada do sistema permitindo determinar a distribuição que mais se ajusta aos dados para entrada no simulador.
- ◆ **Output Analyser**: Realiza a análise estatística dos resultados da simulação.
- ◆ **Arena Viewer**: Visualizador da simulação.

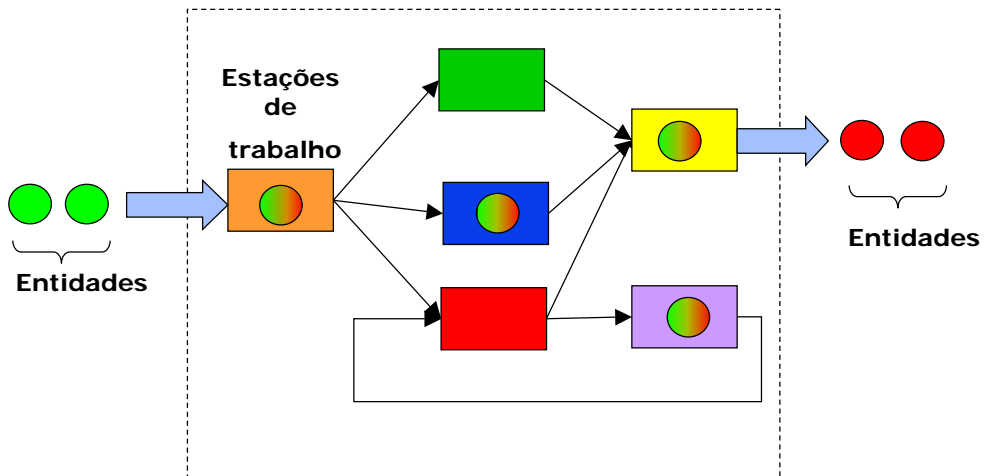
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## *Elementos da Modelagem*

- ◆ **Entidades:** são as pessoas, transações ou tarefas que se movem ao longo do sistema.
- ◆ **Estações de trabalho:** onde será realizado algum serviço.
- ◆ **Fluxo:** caminhos que a entidade irá percorrer ao longo de estações.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

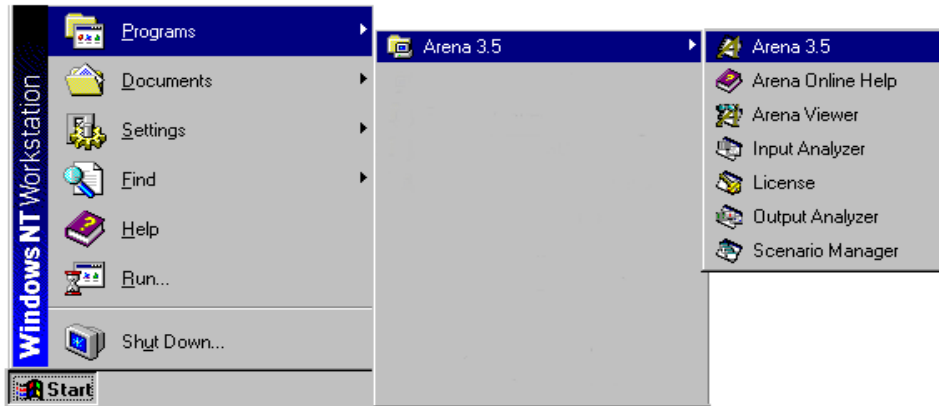
## *Elementos em uma modelagem*



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Primeiros passos

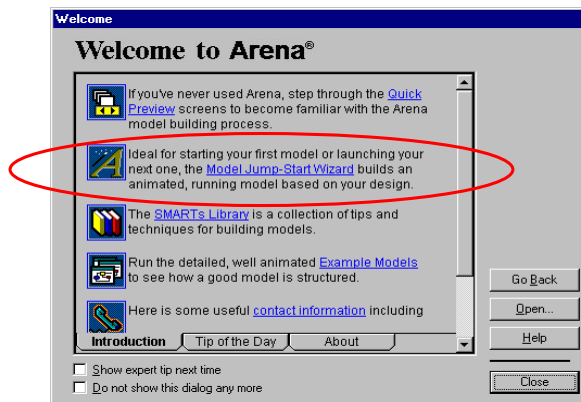
### ◆ Executar o **Arena**



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Primeiros passos

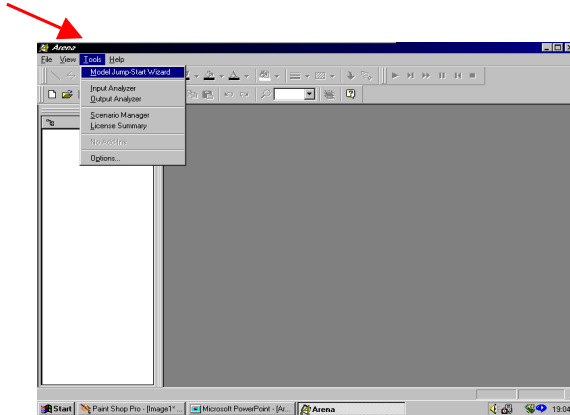
### ◆ Selecionar **File** e **New** para criar um novo modelo ou então selecionar **Model Jump-Start Wizard** que constrói um modelo simples.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## *Primeiros passos*

- ◆ O **Model Jump-Start Wizard** pode ser ativado não apenas da tela inicial mas também através da opção de menu **Tools**.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## *Primeiros passos*

- ◆ O **Model Jump-Start Wizard** permite a criação de modelos com uma ou mais estações de trabalho encadeadas.
- ◆ Pode-se escolher se a entidade será uma pessoa, documento, parte ou geral e neste caso serão representadas por símbolos diferentes.
- ◆ A estação de trabalho também pode ser uma pessoa, máquina ou geral.
- ◆ Note que o Arena não aceita acentuação da língua portuguesa.

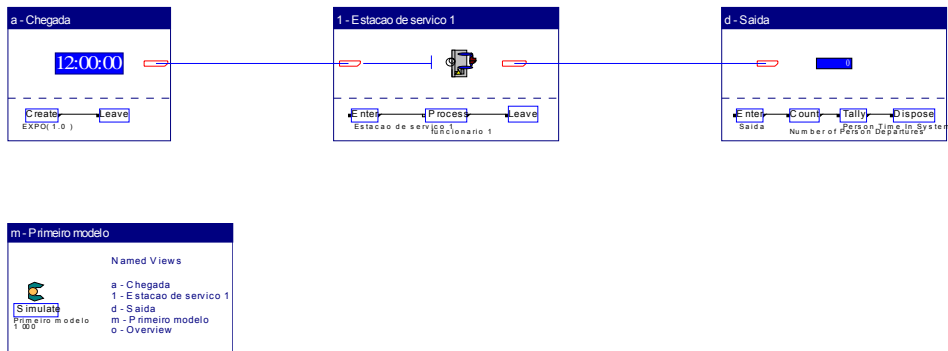
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Primeiros passos

- ◆ Este é um modelo gerado pelo Wizard.

### Primeiro modelo

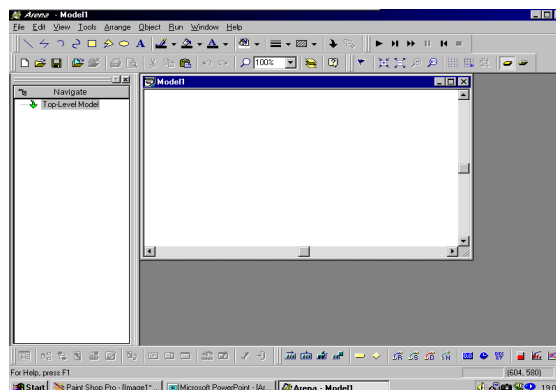
Graça Bressan



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Criação de um Modelo

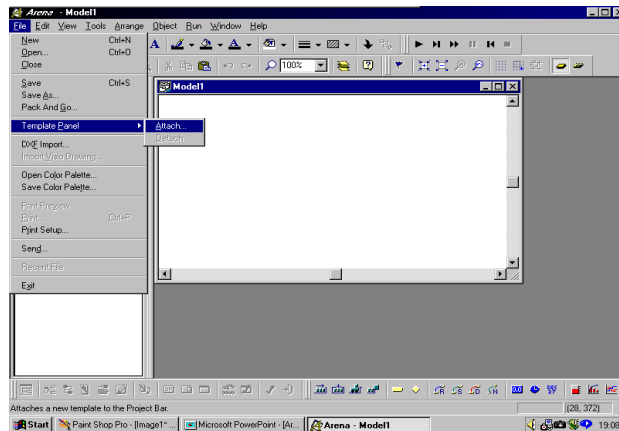
- ◆ Se você seleccionar **File** e **New** em lugar de utilizar o Wizard, o Arena abrirá uma janela em branco para você editar o modelo.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Criação de um Modelo

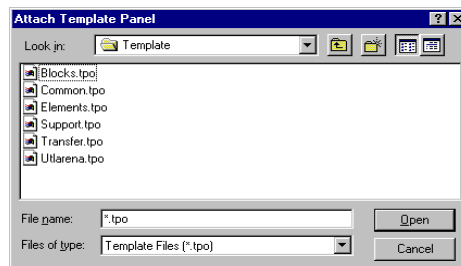
- ◆ Neste caso você deverá seleccionar no menu **File**, **Template Panel** e **Attach**.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Criação de um Modelo

- ◆ Após seleccionar **Attach** será aberta uma janela para seleccionar um arquivo de template. Estes arquivos estarão no diretório C:/Arena/Templates, supondo que o arena está instalado em C:/Arena. Selecione o arquivo **common.tpo** que contém um painel de módulos básicos (**templates**) a serem utilizados.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

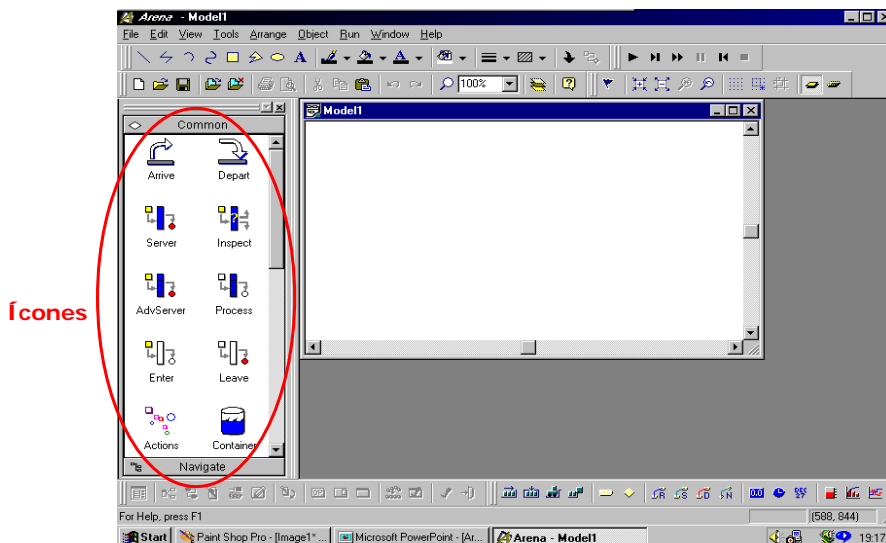


## *Criação de um Modelo*

- ◆ Feito isso, aparecerá na parte esquerda da tela uma caixa de ferramentas com ícones que correspondem aos módulos básicos para criação de modelos. Existem outros arquivos contendo outros painéis, entre os quais o [support.tpo](#).

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados





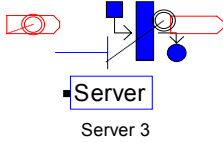
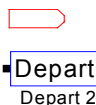
## *Janela de Criação de um Modelo*



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados


## Criação de um Modelo

- ◆ Para inserir um módulo no modelo clique no ícone desejado e arraste-o para a janela de edição do modelo:

<b>Ícone</b>	 Arrive	 Server	 Depart
<b>Módulo inserido</b>	 Arrive Arrive 2	 Server Server 3	 Depart Depart 2

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Criação de um Modelo

- ◆ Para definir os parâmetros do módulo, posicionar o cursor no retângulo que contém o nome do módulo e dar um duplo clique sobre o mesmo.
- ◆ Uma das formas de conectar módulos é através do ícone  que fica no menu na parte superior da janela principal.
- ◆ Clique neste ícone e, em seguida, clique no triângulo negro do módulo inicial e depois dê um clique duplo no retângulo negro do módulo final.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

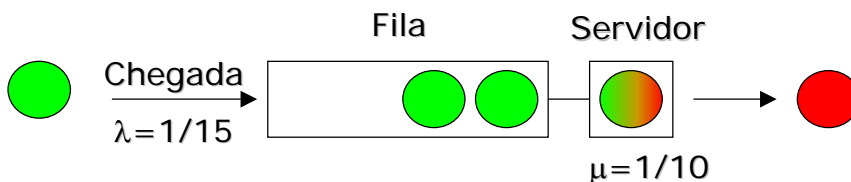
## Exemplo 1: Modelo de Fila Única

- ◆ Um pequeno supermercado possui uma única caixa sendo que os clientes chegam à fila do caixa em intervalos de tempo cuja média é 15 minutos. O tempo médio de atendimento de um cliente pelo caixa é de 10 minutos.
- ◆ Se as distribuições dos intervalos de tempo de chegada e atendimento forem exponenciais então o sistema será denominado M/M/1.
  - Intervalo de chegada exponencial com média 15 min.
  - Fila de tamanho infinito.
  - Tempo médio de atendimento de 10 min.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 1: Modelo de fila M/M/1

- ◆ Se o intervalo de chegada é 15 minutos por cliente e se o intervalo de atendimento for 10 minutos por cliente então a taxa de chegada é  $\lambda = 1/15$  clientes/minuto e a taxa de atendimento é  $\mu = 1/10$  clientes/minuto.
- ◆ A representação deste modelo é:



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## *Construção do modelo M/M/1*

- ◆ Na construção deste primeiro modelo usaremos os seguintes elementos do Arena:

- **Arrive**
- **Server**
- **Depart**
- **Simulate**
- **Connect**

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## *Construção do modelo M/M/1*

- ◆ O primeiro módulo a ser incluído é o **Arrive** ao selecionar o ícone



- ◆ O **Arrive** tem como objetivo gerar entidades segundo uma distribuição especificada.
- ◆ O módulo gerado é



Arrive  
Arrive 1

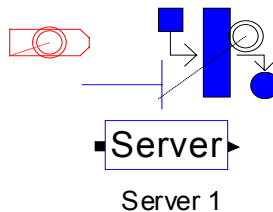
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do modelo M/M/1

- ◆ Vamos agora incluir o módulo **Server** através do ícone



- ◆ O **Server** representa um sistema com uma fila e um ou mais servidores, podendo-se especificar uma distribuição para o tempo de atendimento.
- ◆ O módulo gerado é



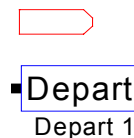
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do modelo M/M/1

- ◆ Este modelo será finalizado com o módulo **Depart** que define a saída do sistema. Este módulo é gerado ao selecionar o ícone



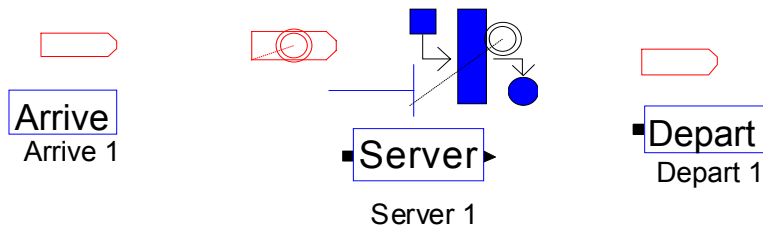
- ◆ O módulo gerado é



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do modelo M/M/1

- ◆ O modelo criado até este ponto é



- ◆ Agora deve ser feita a conexão dos módulos e definição de seus parâmetros.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do modelo M/M/1

- ◆ Após dois cliques no retângulo do módulo **Arrive** irá aparecer a janela de opções que permite definir os parâmetros do módulo.

- ◆ Os parâmetros que iremos definir são:

- Em **Enter Data**, iremos substituir o nome **Arrive 1** por **Chegada**.
- **Time Between**: iremos escolher distribuição exponencial com média 15 (min).
- Em **Leave Data** iremos selecionar a opção **Connect**.

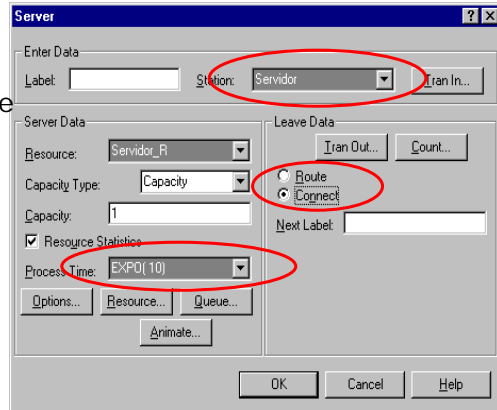
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do modelo M/M/1

- ◆ Após dois cliques no retângulo do módulo **Server** irá aparecer a janela de opções que permite definir os parâmetros do módulo.

Os parâmetros que iremos definir são:

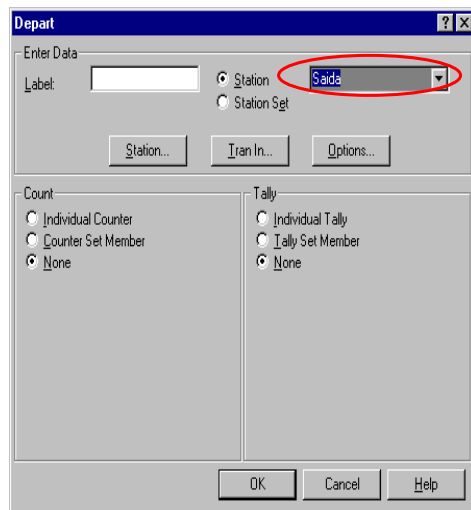
- ◆ Em **Enter Data**, onde aparece **Server 1** iremos colocar **Servidor**. Note que em **Resource** aparecerá **Servidor\_R**.
- ◆ **Process Time**: definirmos distribuição exponencial com média 10(min).
- ◆ Em **Leave Data** iremos selecionar a opção **Connect**.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados


## Construção do modelo M/M/1

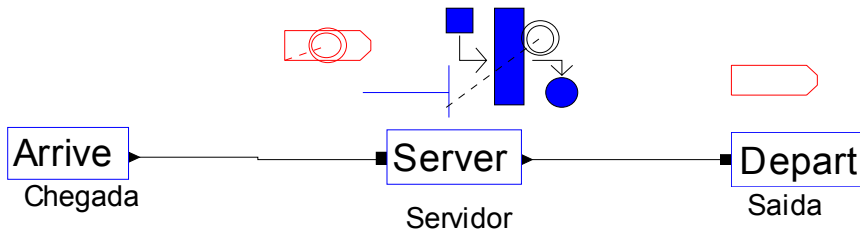
- ◆ Finalmente iremos dar dois cliques no retângulo do módulo **Depart** e irá aparecer a janela para definição de parâmetros.
- ◆ Nesta janela em **Enter Data**, onde aparece **Depart 1** iremos colocar **Saída**.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do modelo M/M/1

- ◆ Para definir as conexões clicamos no ícone  e nos dois módulos, um clique simples no triângulo negro e um clique duplo no quadrado negro, dos módulos a serem conectados. O modelo resultante será



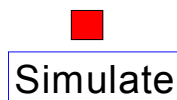
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do modelo M/M/1

- ◆ Para executarmos a simulação teremos ainda que inserir o módulo **Simulate** e definir os parâmetros da simulação.



- ◆ O módulo inserido será



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados



## Construção do modelo M/M/1

- ◆ Finalmente iremos dar dois cliques no retângulo do módulo **Simulate** e irá aparecer a janela para definição de parâmetros.
- ◆ Definimos:
- ◆ **Title:** MM1 que é o nome do modelo (não aceita "/" )
- ◆ **Analyst e Date:** Nome do autor do modelo e data de sua criação.
- ◆ **Length of Replication:** 1000 min que é o tempo total da simulação.

**Simulate**

Project

Title: MM1

Analyst: GB

Date: 12 07 2000

Replicate

Number of Replications: 1

Beginning Time: 0.0

Length of Replication: 1000

Terminating Condition:

Between Replications...

☒ Initialize System

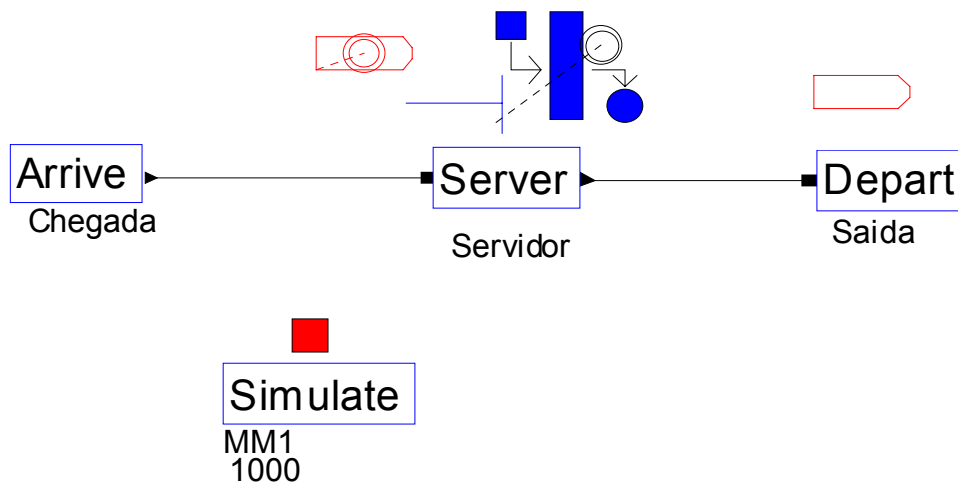
☒ Initialize Statistics

Warm-Up Period:

OK Cancel Help

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Modelo M/M/1



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Execução da Simulação do Modelo M/M/1***

- ◆ Para executar a simulação basta clicar nas opções do menu **Run** e **Go** e ao final da simulação será emitido um relatório.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Relatório da Simulação do modelo M/M/1***

ARENA Simulation Results  
Systems Modeling <user unknown> - License #9400000

Summary for Replication 1 of 1

Project: MM1                                      Run execution date : 12/ 7/2000  
Analyst: GB                                        Model revision date: 12/ 7/2000

Replication ended at time : 1000.0

### **TALLY VARIABLES**

Identifier	Average	Half Width	Minimum	Maximum	Observations
------------	---------	------------	---------	---------	--------------

Servidor_R_Q Queue Tim	21.513	(Insuf)	.00000	53.517	68
------------------------	--------	---------	--------	--------	----

### **DISCRETE-CHANGE VARIABLES**

Identifier	Average	Half Width	Minimum	Maximum	Final Value
------------	---------	------------	---------	---------	-------------

Servidor_R Busy	.79548	(Insuf)	.00000	1.0000	1.0000
Servidor_R Available	1.0000	(Insuf)	1.0000	1.0000	1.0000
# in Servidor_R_Q	1.5160	(Insuf)	.00000	5.0000	3.0000

Simulation run time: 0.00 minutes.  
Simulation run complete.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## **Relatório da Simulação do modelo M/M/1**

- ◆ Podemos extrair desse relatório os seguintes resultados que podem ser comparados os resultados analíticos do sistema M/M/1.

Tempo de simulação	1000.0 min
Tempo médio na fila	21.513 min
<del>◆ A fração do tempo que o servidor ficou disponível é 0.95 (100%) pois não ocorreu falha ou outro evento que tirasse o servidor de operação</del>	<del>0.95 (100%)</del>
Número médio de clientes na fila	1.5160

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## **Exercícios com o Simulador do Sistema M/M/1**

1. Verifique os resultados da simulação por tempos maiores, tais como 5.000, 10.000, etc. Compare os resultados com os valores obtidos através das fórmulas analíticas do sistema M/M/1. Verifique como os valores da simulação se aproximam dos valores teóricos (faça um gráfico da aproximação).
2. Utilize outras distribuições e analise os resultados obtidos. Note que se não for possível obter uma distribuição que se ajuste bem aos dados reais, é possível entrar o vetor de frequências através da escolha da distribuição CONT.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Exercícios com o Simulador do Sistema M/M/1***

3. Para construir o modelo do sistema M/M/m (uma fila e m servidores) basta uma mudança muito simples. Na janela de parâmetros do servidor alterar o parâmetro **capacity** que está definido como 1 para o valor m escolhido. Considerando  $m=3$ , altere o modelo e execute novamente o simulador. Agora, através de execuções do simulador, você pode responder o que é melhor:

- Uma única fila com taxa de chegada  $\lambda$  e 3 servidores com taxa de atendimento  $\mu$  cada um ou,
- 3 filas independentes sendo cada fila com um servidor com taxa de atendimento  $\mu$ , e que cada fila recebe  $1/3$  dos clientes, isto é, a taxa de chegada de cada fila é  $\lambda/3$ .

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Exemplo 2: Linha de Montagem***

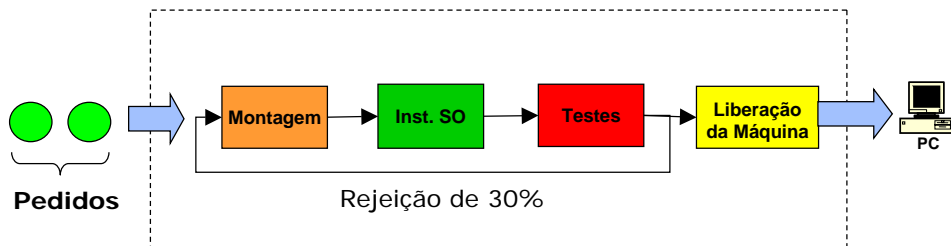
- ◆ Em uma empresa montadora de computadores um computador passa, de forma simplificada, pelas etapas (a unidade de tempo é minutos) :
    1. Chegada de um pedido (distr. Exponencial, média=100);
    2. Montagem de módulos (distr. Triangular, mínimo=60, moda=80 min, máximo=100);
    3. Instalação do sistema operacional (distr. Triangular, mínimo=80, moda=100 min, máximo=120);
    4. Testes de funcionamento (distr. Triangular, mínimo=50, moda=60, máximo=70) com índice de falhas=30% e liberação da máquina se for aprovada nos testes, caso contrário volta à etapa 2.
- Obs.: Considerar que o tempo de deslocamento entre uma estação e a seguinte é de 2 minutos em todas etapas.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 2: Linha de Montagem

### ◆ Desejamos saber:

- O tempo médio para liberação de um computador após o pedido.
- A produção da empresa após 720 minutos de simulação (12 horas).



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do Modelo da Linha de Montagem

### ◆ Na construção deste modelo serão utilizados os seguintes elementos do Arena além de outros já apresentados:

- **Inspect**
- **Route**
- **Atributos de Entidades**
- **Animação de Estações**

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 2: Rotas

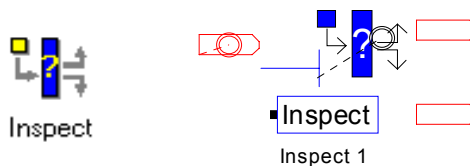
### ◆ Route

- Em lugar de se escolher **connect**, pode-se escolher **route** na definição dos parâmetros de uma estação, que permite especificar o nome da próxima estação (**station**) e o tempo que demora para chegar à próxima estação por esta rota (**Route Time**). Todos os módulos, em geral, permitem a definição de rota.
- Note que no gráfico a rota não aparece visível, ao contrário da conexão.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 2: Módulo Inspect

- ◆ Permite a especificação de probabilidade de falha e qual será a próxima estação neste caso. É utilizado para definir estações de teste ou inspeção.

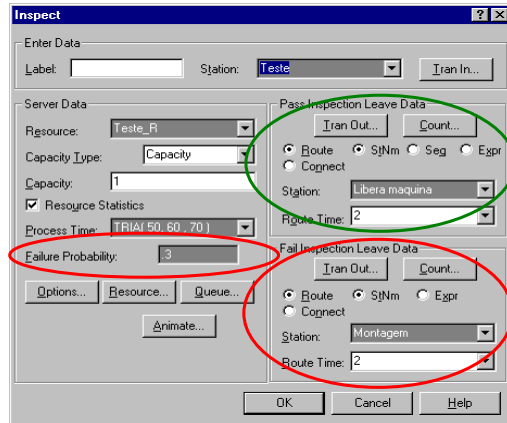


© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 2: Módulo Inspect

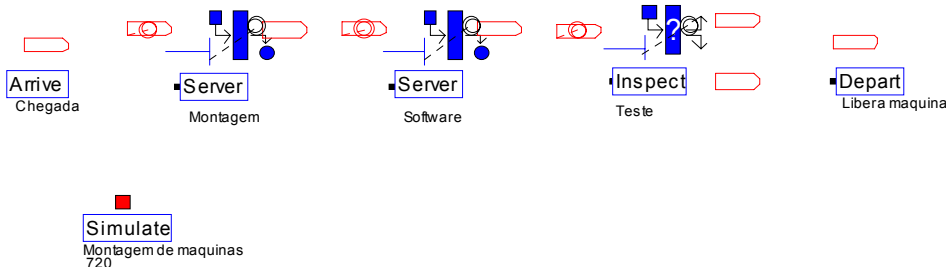
◆ Parâmetros além dos já utilizados nas outras estações:

- **Failure probability** onde se define a probabilidade de falha;
- Definição de rota ou conexão, nos casos de sucesso ou falha.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 2: Modelo da Linha de Montagem



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 2: Atributo de Entidade

- ◆ Notem que se executarmos a simulação não obteremos os resultados solicitados:
  - O tempo médio para liberação de um computador após o pedido.
  - A produção da empresa após 720 minutos de simulação (12 horas).
- ◆ O cálculo desses resultados exige a introdução do conceito de **atributo de entidade** que é um valor que acompanha a entidade ao longo das estações.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 2: Atributo de Entidade

- ◆ Para calcular o tempo entre a chegada do pedido e entrega do produto, primeiramente inserimos no módulo **Arrive**, o atributo **Tempo de chegada** (um nome de nossa escolha) no campo **Mark Time Attribute**.

The screenshot shows the 'Arrive' dialog box with the following settings:

- Enter Data:** Station: **Chegada**
- Arrival Data:** Batch Size: 1, Time Between: EXP(100), Mark Time Attribute: **Tempo de chega** (highlighted with a red circle)
- Leave Data:** Station: Montagem, Route Time: 2

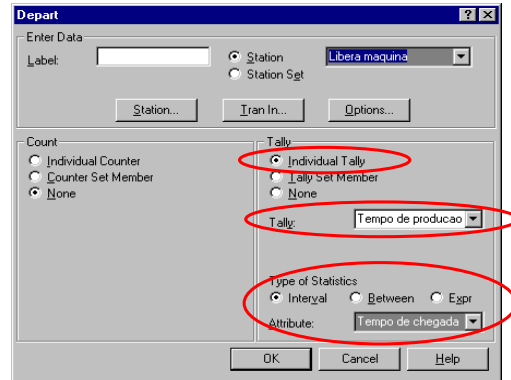
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados



## Exemplo 2: Atributo de Entidade

Em seguida alteramos os seguintes parâmetros de **Tally** do módulo **Depart**:

- Seleccionamos **Individual Tally**.
- Preenchemos o campo **Tally** com um nome que aparecerá no relatório, no caso foi escolhido **Tempo de produção**.
- No Campo **type of statistics** seleccionamos **Interval** e em **Attribute** escolhemos o atributo **Tempo de chegada**.



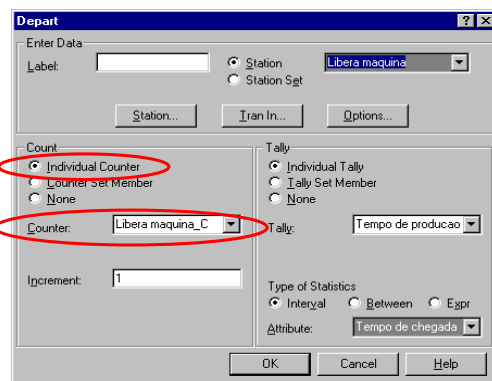
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 2: Atributo de Entidade

Para determinar o número de máquinas produzidas, alteramos os seguintes parâmetros de **Counter** no módulo **Depart**:

- Seleccionamos **Individual Counter**.
- Seleccionamos **Libera maquina\_C** no campo **Counter**.

Notar que apareceu um contador no módulo **Depart** do modelo. Se a simulação durar um tempo maior pode-se observar o contador sendo incrementado.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 2: Atributo de Entidade

ARENA Simulation Results  
Systems Modeling <user unknown> - License #9400000

Summary for Replication 1 of 1

Project: Montagem de maqu                      Run execution date : 18/ 7/2000  
Analyst: GB                                      Model revision date: 14/ 7/2000

Replication ended at time                      : 720.0

### TALLY VARIABLES

Identifier	Average	Half Width	Minimum	Maximum	Observations
Teste_R_Q Queue Time	.00000	(Insuf)	.00000	.00000	5
Software_R_Q Queue Tim	18.349	(Insuf)	.00000	63.428	6
Tempo de producao	248.72	(Insuf)	241.38	256.54	3
Montagem_R_Q Queue Tim	31.283	(Insuf)	.00000	66.562	8

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 2: Atributo de Entidade

### DISCRETE-CHANGE VARIABLES

Identifier	Average	Half Width	Minimum	Maximum	Final Value
Software_R Busy	.74968	(Insuf)	.00000	1.0000	1.0000
Teste_R Busy	.37060	(Insuf)	.00000	1.0000	1.0000
# in Software_R_Q	.17163	(Insuf)	.00000	1.0000	1.0000
# in Montagem_R_Q	.41341	(Insuf)	.00000	2.0000	1.0000
# in Teste_R_Q	.00000	(Insuf)	.00000	.00000	.00000
Montagem_R Available	1.0000	(Insuf)	1.0000	1.0000	1.0000
Montagem_R Busy	.79761	(Insuf)	.00000	1.0000	1.0000
Teste_R Available	1.0000	(Insuf)	1.0000	1.0000	1.0000
Software_R Available	1.0000	(Insuf)	1.0000	1.0000	1.0000

### COUNTERS

Identifier	Count	Limit
Libera maquina_C	3	Infinite

Simulation run time: 0.00 minutes.  
Simulation run complete.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exercício

- ◆ Altere o modelo de forma a incluir uma etapa 5 ao final em que é feita a identificação do tipo do problema e volta à etapa 2 ou 3, conforme o caso:
  - **Etapa 5:** Identificação do problema, caso ocorra algum, e retorno à etapa 2 ou 3 com probabilidades 40%, se falha de HW, e 60%, se falha de SW (distribuição triangular mínimo=20, moda=30 min, máximo=60).

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

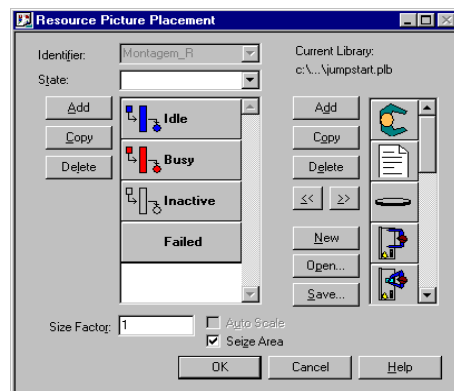
## Alteração das Figuras da Animação

- ◆ Pode-se trocar as figuras que representam as estações e entidades. Para isso basta clicar duas vezes sobre o desenho a ser mudado.

- Por exemplo, podemos mudar o símbolo de um servidor dando um duplo clique sobre o desenho.



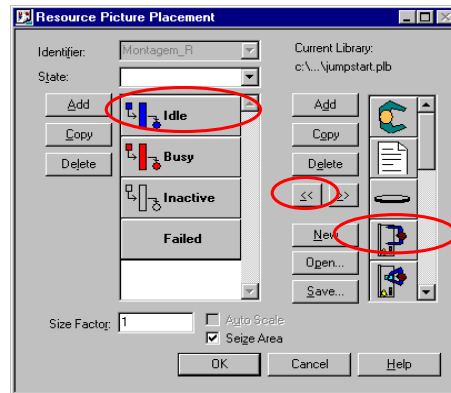
- Aparecerá a janela ao lado onde podem ser escolhidas novas figuras.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Alteração das Figuras da Animação

- Para mudar a figura **idle**, por exemplo, clicar nesta figura, em seguida na figura da biblioteca que queremos utilizar e por último no botão <<.

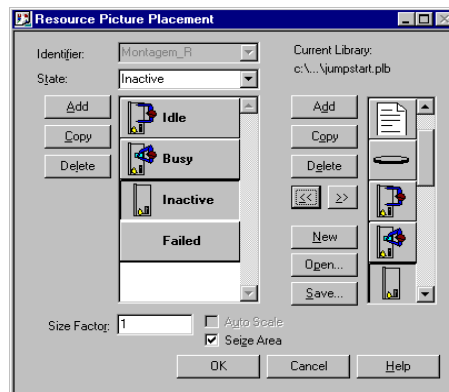


© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Alteração das Figuras da Animação

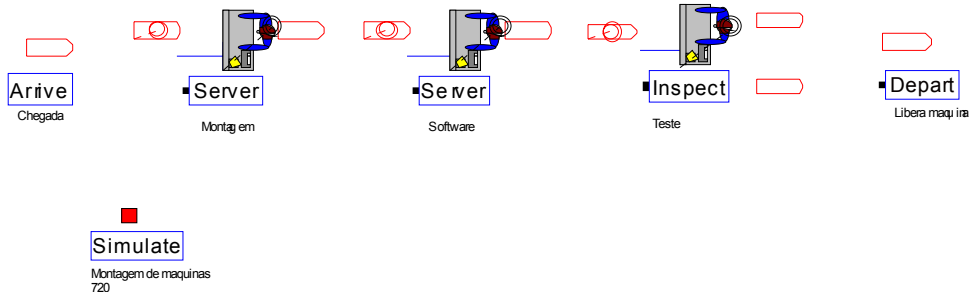
No exemplo mudamos os símbolos do servidor nos estados **Idle**, **Busy** e **Inactive**.

Pode-se escolher símbolos contidos nas bibliotecas disponíveis, com extensão .plb, ou pode-se criar novas bibliotecas e novos símbolos. Os que foram utilizados estão na biblioteca **JumpStart.plb** existente no diretório do **Arena**.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Alteração das Figuras da Animação



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 3: Linha de Montagem

◆ Em uma indústria de automóveis são fabricados três modelos de automóveis com mesma base mecânica:

- Carro A: Sedã popular (pedidos de 50 carros/mês \*)



- Carro B: Sedã luxo (pedidos de 25 carros/mês \*)



- Carro C: Picape (pedidos de 25 carros/mês \*)



\* Considerar: distribuições exponenciais e mês com 200 horas.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

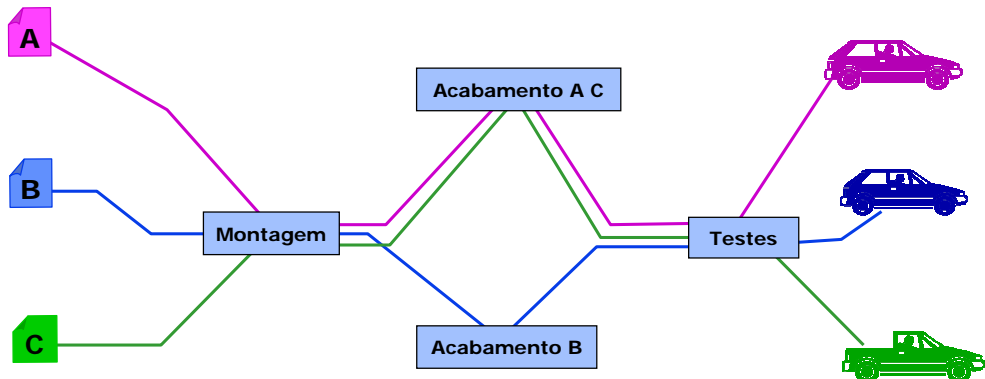
## Exemplo 3: Linha de Montagem

◆ A produção é realizada nas seguintes estações:

- Montagem: Montagem dos carros A, B e C.
- Acabamento A C: Acabamento de A e C.
- Acabamento B: Acabamento de B.
- Testes: Testes de A, B e C.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 3: Linha de Montagem



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

### Exemplo 3: Linha de Montagem

- ◆ Os tempos de atendimentos tem distribuição Triangular com os seguintes parâmetros:

Estação	Mínimo (horas)	Moda (horas)	Máximo (horas)
Montagem	2,8	3	3,2
Acabamento A C	2,5	2,6	2,8
Acabamento B	2,8	3	3,2
Testes	0,4	0,5	0,6

- ◆ A estação de montagem utiliza duas linhas operando em paralelo (capacidade 2).

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

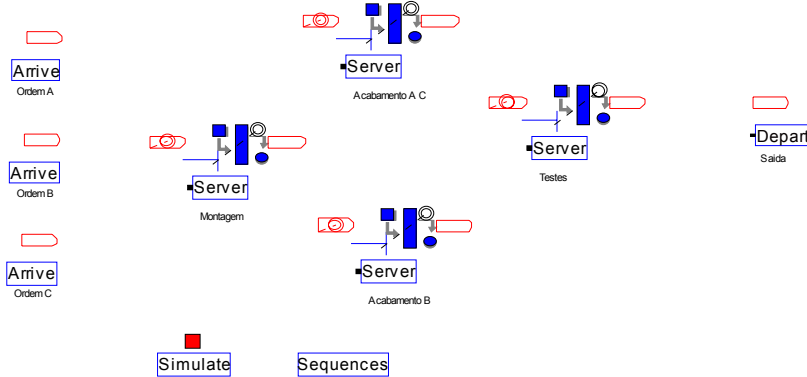
### Construção do Modelo de Linha de Montagem

- ◆ Na construção deste modelo serão utilizados os seguintes novos elementos do Arena:
  - **Sequence**
  - **Animação de Entidades**
- ◆ O módulo **sequence** é o que permitirá a definição de diferentes fluxos de entidades pelas estações para que possamos definir seqüências de produção diferentes para os carros A, B e C.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do Modelo de Linha de Montagem

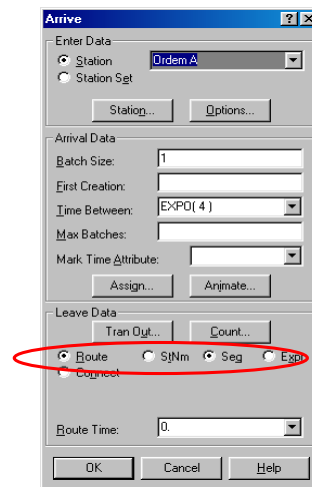
- ◆ A construção foi feita com os módulos **Arrive**, **Server**, **Depart**, **Simulate**, como nos exemplos anteriores, tendo-se incluído o módulo **sequence**.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do Modelo de Linha de Montagem

- ◆ Na definição dos parâmetros dos módulos ocorrerão as primeiras diferenças:
  - No bloco **Leave Data** onde é feita a definição de **route**, em lugar da seleção **StNm** como estava sendo feita nos exemplos anteriores, será selecionada **Seg**, como mostra a janela ao lado, para que a rota seja determinada por uma sequência.
- ◆ Esta definição deverá ser feita em todos blocos **Leave Data** de todas estações **Arrive** e **Server**.

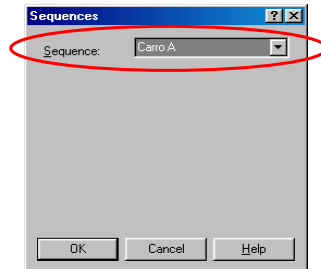
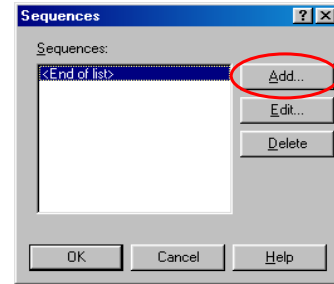


© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados



## Construção do Modelo de Linha de Montagem

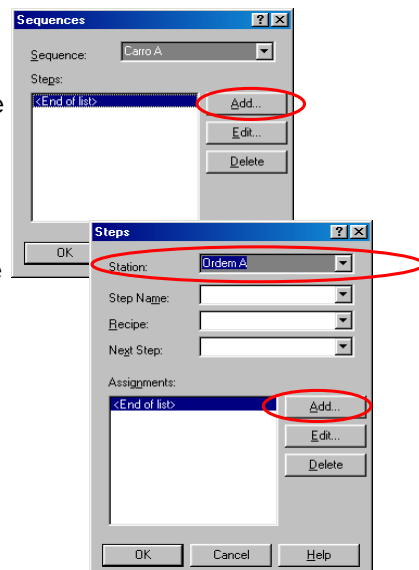
- ◆ Iremos definir três seqüências diferentes que chamaremos **Carro A**, **Carro B** e **Carro C**. Para isso, clicamos duas vezes no módulo **sequence** para abrir a janela de parâmetros.
  - Na primeira janela que aparece, clicamos em **Add** e será aberta uma segunda janela.
  - Nesta segunda janela existe uma caixa de texto que preenchemos com o nome da seqüência que queremos definir, neste caso **Carro A**. Ao clicarmos **OK** aparecerá uma mensagem **too few entries...** pois não definimos a seqüência de estações.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do Modelo de Linha de Montagem

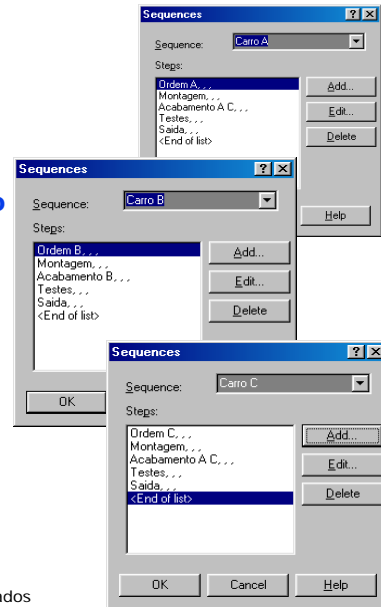
- ◆ Na janela que aparece a seguir definimos a seqüência de estações que corresponde a **Carro A**, para isso clicamos no botão **Add**.
- ◆ Na janela que será aberta definimos as estações da seqüência: selecionamos uma estação na caixa de seleção **Station** e clicamos no botão **Add** e em seguida **OK**.
- ◆ Fazemos o mesmo para todas as estações que fazem parte da seqüência **Carro A**.
- ◆ Existem outros parâmetros que não serão utilizados no momento.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do Modelo de Linha de Montagem

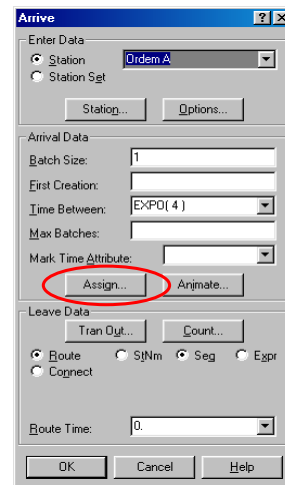
- ◆ A janela ao lado mostra o resultado final da definição da sequência **Carro A**.
- ◆ Repetimos o mesmo processo para definir as sequências **Carro B** e **Carro C**.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do Modelo de Linha de Montagem

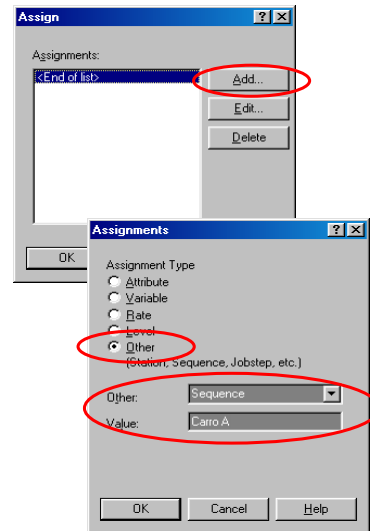
- ◆ Após a definição das sequências, ainda está faltando associar as sequências com as estações **Arrive**.
- ◆ Para isso, clicamos em uma estação **Arrive** e, na janela de parâmetros, clicamos o botão **Assign** nesta janela.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do Modelo de Linha de Montagem

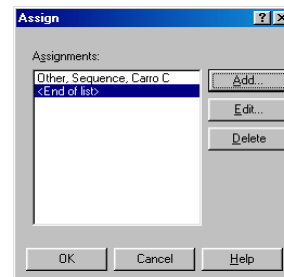
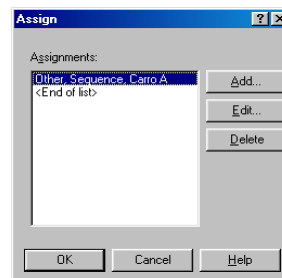
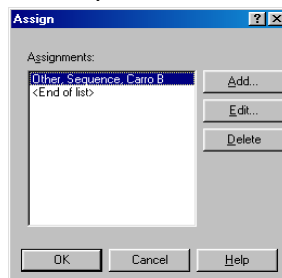
- ◆ Na janela do **Assign** deve-se clicar no botão **Add** que abrirá nova janela.
- ◆ Nesta nova janela selecionar **Other** e nas caixas de textos que irão aparecer, selecionar:
  - **sequence**
  - Na caixa de texto do **value** colocar o nome da sequência, por exemplo, **Carro A**.
- ◆ Desta forma, em cada comando **Arrive** definiremos a sequência correspondentes cada tipo de carro.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Construção do Modelo de Linha de Montagem

- ◆ Desta forma, em cada comando **Arrive** definiremos a sequência correspondentes a cada tipo de carro.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

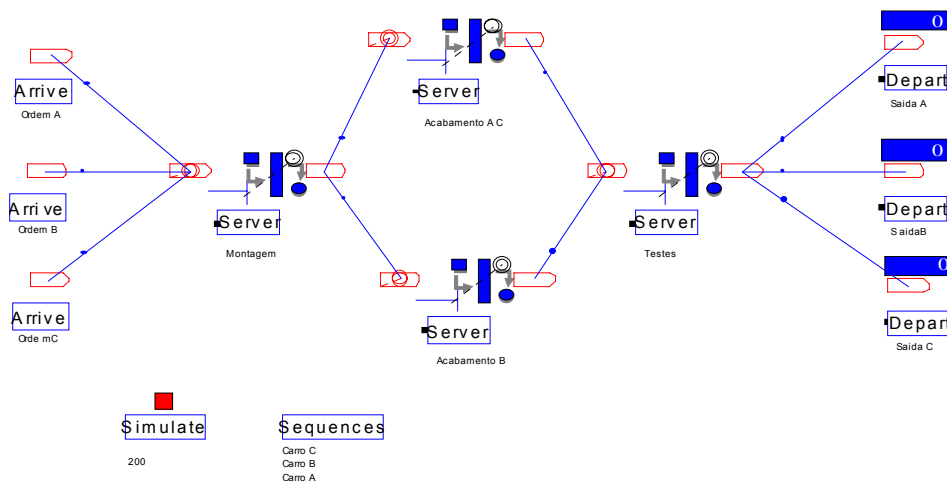
## Construção do Modelo de Linha de Montagem

- ◆ Para obter valores do número de carros e o tempo de produção de cada tipo foram introduzidas estações **depart** para cada tipo de carro.
- ◆ Também foram definidas as ligações entre as estações através animação de rotas



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Modelo de Linha de Montagem



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

# Modelo de Linha de Montagem

## ARENA Simulation Results Summary for Replication 1 of 1

Project: Run execution date : 7/22/2000  
Analyst: Model revision date: 7/22/2000

Replication ended at time : 200.0

### TALLY VARIABLES

Identifier	Average	Half Width	Minimum	Maximum	Observations
Acabamento A C_R_Q Que	3.6245	(Insuf)	.00000	13.560	64
Acabamento B_R_Q Queue	.84485	(Insuf)	.00000	4.8273	29
Montagem_R_Q Queue Tim	2.5936	(Insuf)	.00000	12.539	100
Tempo de A	12.820	(Insuf)	5.9785	24.964	38
Testes_R_Q Queue Time	.03041	(Insuf)	.00000	.44263	92
Tempo de B	9.4845	(Insuf)	6.1623	17.766	29
Tempo de C	10.888	(Insuf)	6.0408	24.675	25

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

# Modelo de Linha de Montagem

### DISCRETE-CHANGE VARIABLES

Identifier	Average	Half Width	Minimum	Maximum	Final Value
# in Testes_R_Q	.01399	(Insuf)	.00000	1.0000	.00000
Acabamento B_R Availab	1.0000	(Insuf)	1.0000	1.0000	1.0000
# in Acabamento B_R_Q	.12250	(Insuf)	.00000	2.0000	.00000
Acabamento B_R Busy	.43600	(Insuf)	.00000	1.0000	.00000
# in Acabamento A C_R_	1.2838	(Insuf)	.00000	6.0000	5.0000
Testes_R Available	1.0000	(Insuf)	1.0000	1.0000	1.0000
Acabamento A C_R Busy	.83062	(Insuf)	.00000	1.0000	1.0000
Testes_R Busy	.23026	(Insuf)	.00000	1.0000	.00000
# in Montagem_R_Q	1.3155	(Insuf)	.00000	9.0000	3.0000
Acabamento A C_R Avail	1.0000	(Insuf)	1.0000	1.0000	1.0000
Montagem_R Available	2.0000	(Insuf)	2.0000	2.0000	2.0000
Montagem_R Busy	1.4876	(Insuf)	.00000	2.0000	2.0000

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## *Modelo de Linha de Montagem*

### COUNTERS

Identifier	Count	Limit
Contador de A	38	Infinite
Contador de B	29	Infinite
Contador de C	25	Infinite

Simulation run time: 0.00 minutes.  
Simulation run complete.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

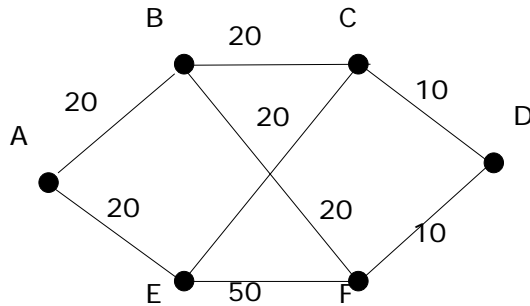
## *Exercícios*

1. Analise o comportamento do sistema e verifique se as estações estão com capacidade inferior à necessária e quanto é necessário para atender à demanda.
2. Construa um refinamento deste modelo em que pedidos de carros são agrupados em **batch** de 10 carros antes de entrarem em produção. Os carros A e B são agrupados em um único **batch** enquanto que os carros de tipo C são agrupado em outro **batch**. Entre a produção de um **batch** de carros A ou B e um **batch** de carros C é necessário um tempo de transição de 0,5 hora para preparação das linhas de montagem (usar módulos **Batch** e **Split**).

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 4: Roteadores

- ◆ Dada a rede de comutação de pacotes com nós A, B, C, D, E e F da figura, com tráfego fim a fim definido por uma dada matriz de roteamento. Cada enlace é full duplex e tem a capacidade indicada na figura em cada sentido em Kbps.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 4: Roteadores

- ◆ O tráfego é simétrico, de acordo com a tabela abaixo, e têm distribuição exponencial. O tamanho dos pacotes tem distribuição exponencial com média de 800 bits/pacote.

	Destino						
		A	B	C	D	E	F
Origem	A		9	4	1	7	4
	B	9		8	3	2	4
	C	4	8		3	3	2
	D	1	3	3		3	4
	E	7	2	3	3		5
	F	4	4	2	4	5	

Matriz de tráfego  $v_{ij}$  (em pacotes por segundo)

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

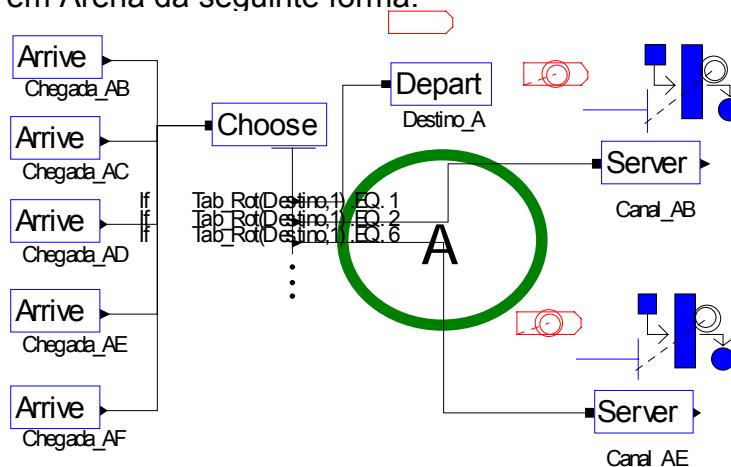
## Exemplo 4: Roteadores

- ◆ Deve ser simulado o comportamento desta rede utilizando tabela fixa de roteamento devendo-se determinar o atraso médio total.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 4: Roteadores

- ◆ A simulação de um nó, no caso foi escolhido o nó A, pode ser feita em Arena da seguinte forma:

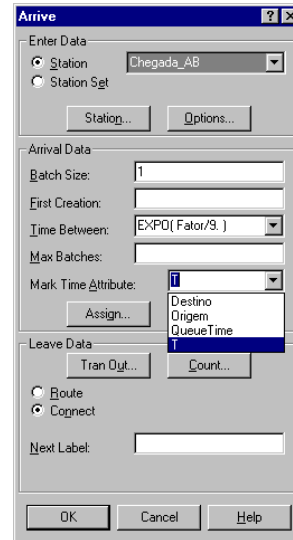


© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados



## Exemplo 4: Roteadores

- ◆ Cada bloco **Arrive** define uma chegada ao nó A com destino a um dos demais nós
- ◆ Para cada **Arrive** definem-se atributos indicando o nó origem e o nó destino da entidade e o instante T em que a entidade foi gerada.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

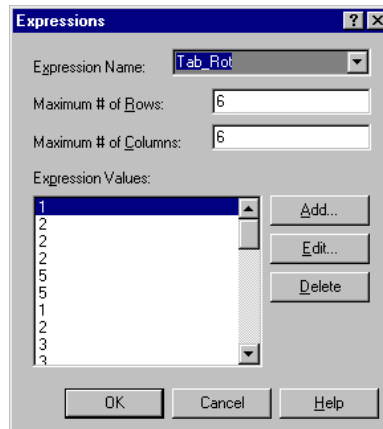
## Exemplo 4: Roteadores

- ◆ A tabela de roteamento é definida em um bloco **Expressions**.

Expressions

Fator  
Tab\_Rot

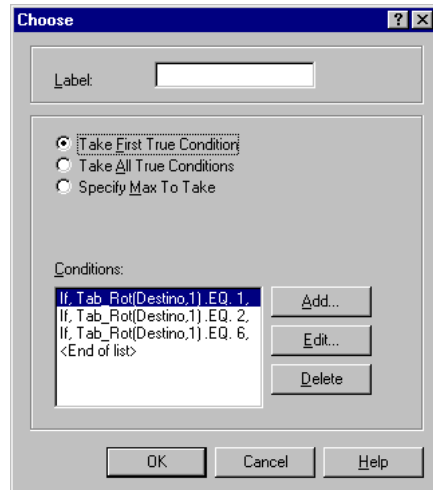
- ◆ Neste bloco foi definida uma matriz 6x6.
- ◆ Deve ser observado que os valores são introduzidos na forma de uma única coluna que contém as colunas da tabela de roteamento.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 4: Roteadores

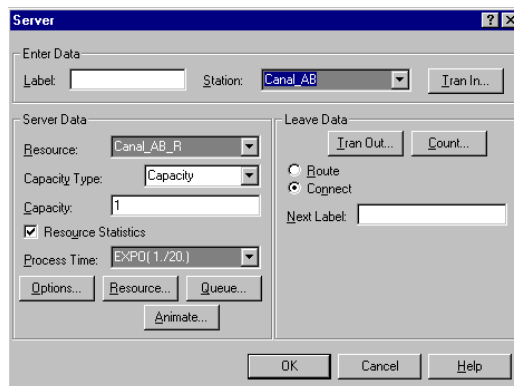
- ◆ A decisão para qual direção tomar, baseada na tabela de roteamento, é feita através do bloco **Choose**, e consultando o atributo de destino da entidade.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 4: Roteadores

- ◆ Cada canal é simulado por um bloco **Server** onde o tempo médio de transmissão de uma mensagem é definido no campo **Process Time**.



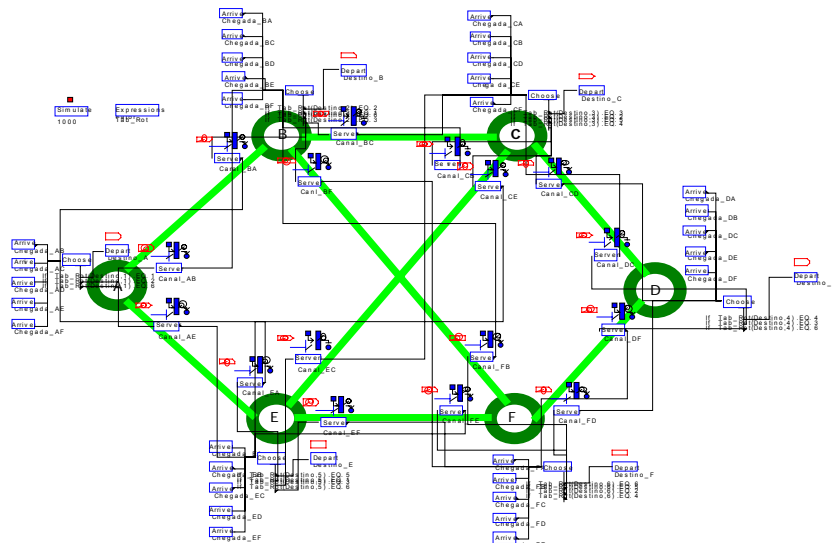
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 4: Roteadores

- ◆ A figura do próximo slide mostra a rede completa.
- ◆ Esta solução está no limite da versão acadêmica quanto ao número de blocos possíveis.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Exemplo 4: Roteadores



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Exercício: Servidor de e-mail***

- ◆ Em um sistema de e-mail, o servidor recebe mensagens, para um único destinatário ou para listas de destinatários.
- ◆ As mensagens chegam em intervalos com distribuição exponencial com média **2 min**.
- ◆ Pode ocorrer falha irrecuperável no envio na primeira tentativa com probabilidade **2%**.
- ◆ Se ocorre falha recuperável (servidor destino estava fora do ar) serão feitas novas tentativas de envio. Em uma tentativa **i** de envio, a mensagem esperará um tempo  **$T = 400 * i * 2 \text{ min}$**  sendo **T** o intervalo entre uma tentativa e a próxima.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Exercício: Servidor de e-mail***

- ◆ **60 %** das mensagens são para um único destinatário enquanto que **40 %** são para listas com número de destinatários com distribuição exponencial com média 5 e mínimo 1 ( **$1 + \text{EXPO}(5)$** ).
- ◆ O tempo de processamento de uma mensagem em caso de sucesso tem distribuição exponencial com média **1 min**. Parte deste tempo é para localizar o endereço do destinatário e o restante é o tempo de transmissão. Em caso de falha a média é **0,5 min** com distribuição exponencial.

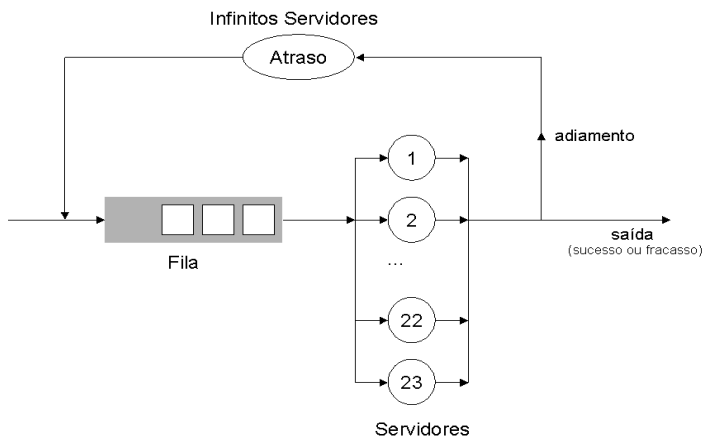
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Exercício: Servidor de e-mail***

- ◆ A probabilidade de falha no envio em cada tentativa é:
  - 1a. tentativa = 5 % de falhas sendo 2% irrecuperáveis
  - 2a. tentativa = 60 % de falhas
  - 3a. tentativa = 80 % de falhas
  - 4a. tentativa = 90 % de falhas
  - 5a. Tentativa = 95% de falhas
- ◆ O número máximo de tentativas é **5**.
- ◆ O servidor de e-mail trabalha com um número processos concorrentes sendo **23** o número default de processos.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Exercício: Servidor de e-mail***



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Elementos Básicos do Arena***

### **Entidades (entity)**

São os objetos que se movem ao longo do sistema. Podem ser transações, clientes, equipamentos sendo produzidos.

### **Estações (station)**

As entidades se dirigem às estações para a realização de serviços. Nas estações, entrarão em filas de espera até serem atendidas.

**Server**, **AdvServer**, **Enter** e **Station** são estações.

### **Recursos (resource)**

São os elementos necessários à realização de um serviço. Podem pertencer a uma única estação ou serem utilizados por várias estações.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Elementos Básicos do Arena***

### **Filas (queue)**

São módulos onde as entidades esperam enquanto aguardam um recurso.

### **Fluxos**

São os caminhos percorridos pelas entidades ao longo do sistema. Os caminhos são definidos através de especificação de rotas, conexões, rótulos, transportadores e esteiras.

### **Painéis de módulos (Templates)**

São arquivos com extensão .tpo que contém um conjunto de módulos. Common.tpo é o que contém os módulos básicos que temos utilizado.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Elementos Básicos do Arena***

### **Atributos**

São associados às entidades para registrar algum valor específico da entidade, isto é, acompanham cada entidade individualmente. Por exemplo, um atributo pode registrar o instante em que a entidade foi criada e consultando o atributo na saída sabemos o tempo que a entidade ficou no sistema.

### **Variáveis**

São variáveis globais cujos valores podem ser utilizados ou alterados ao longo da simulação. São diferentes de atributos pois variáveis podem registrar valores comuns a várias entidades.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Elementos Básicos do Arena***

### **Animação**

Os elementos do modelo podem ser animados por figuras, permitindo a visualização durante a simulação.

### **Seqüências (sequences)**

Pode-se definir uma rota determinada as entidades seguirem entidades seguirem através da definição da seqüência. Entidades pertencentes a seqüências diferentes poderão seguir rotas diferentes a medida que passam por estações. Cada entidade possui dois atributos associados e que podem ser verificados:

NS: Nome da seqüência à qual pertence

IS: Nome da estação atual onde está.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## *Elementos Básicos do Arena*

### **Transportadores (transporters) e Esteiras (conveyors)**

São módulos que transportam as entidades de uma estação a outra.

### **Depósitos (storage)**

São lugares de armazenamento de entidades, tais como peças produzidas.

### **Conjuntos (sets)**

Recursos de um mesmo tipo podem ser agrupados em conjuntos.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## *Elementos Básicos do Arena*

### **Estatísticas**

São resultados contabilizados pelo Arena durante a simulação e que são apresentados no relatório final.

- ♦ **Counters:** contadores de entidades que estão em um determinado módulo do modelo.
- ♦ **Tallies:** Grandezas resultantes de observação tais como tempo entre duas estações.
- ♦ **Frequencies:** distribuição de frequência de uma determinada grandeza.
- ♦ **Time-persistent:** grandezas que são dependentes do tempo, tais como tamanho de filas.
- ♦ **Outputs:** resultados finais de estatísticas e outros valores coletados.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados



## *Caminho entre Estações*

Os caminhos são definidos através de especificação de rotas, conexões, rótulos, transportadores e esteiras.

- Conexões realizadas através da operação **connect** referência ao rótulo (**label**) do próximo módulo.
- Referência ao nome da próxima estação e recurso de transporte (**route**, **transporters** ou **conveyors**). Na definição de rota pode-se especificar a duração da rota. No caso de **transporters** e **conveyors** também pode-se especificar o tempo de carga.



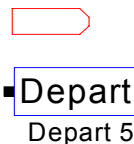
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## *Arrive*



- ◆ Módulo de chegada de entidades ao sistema. Permite definir a distribuição dos intervalos de tempo da chegada de entidades.

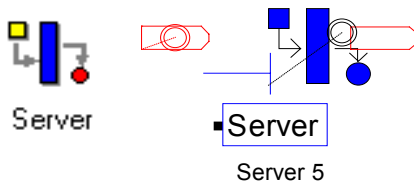
## *Depart*



- ◆ Módulo de saída de entidades do sistema. Neste módulos se especificam estatísticas a serem incluídas no relatório, tais como **Tally** e **Counter**.

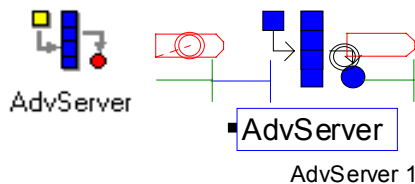
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Server



- ◆ Módulo que inclui uma fila de espera e um ou mais recursos para a realização de serviços.

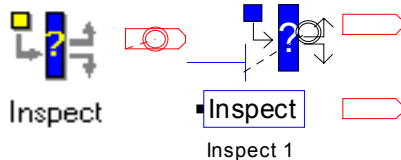
## AdvServer



- ◆ É um servidor com recursos adicionais, permitindo entre outras coisas que o recurso pertença a um conjunto ou seja um **transporter** e pode fazer referência a uma receita (**recipe**).

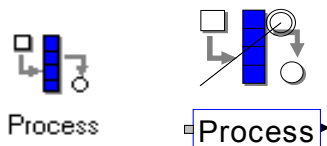
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Inspect



- ◆ Módulo que inclui uma fila de espera e um ou mais servidores e cujo serviço resulta em uma verificação. O resultado da verificação é definido por uma porcentagem de sucesso ou falha e a entidade é encaminhada a uma estação ou outra conforme o resultado.

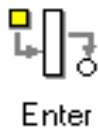
## Process



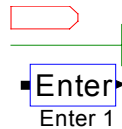
- ◆ Realiza um serviço e para isto exige a alocação de um recurso. Juntamente com o Enter e **Leave** permite compor um módulo equivalente ao **Server** ou **AdvServer**.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Enter



Enter

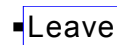


- ◆ Define uma estação ou conjunto de estações onde um processamento irá ocorrer. Pode liberar transportador ou definir uma receita. Tem uma fila associada.

## Leave



Leave



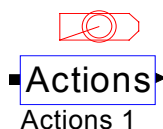
- ◆ É utilizado para liberar um recurso e encaminhar a entidade para outra estação. Permite a espera de transportador ou esteira e especificar um tempo de carga.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Actions



Actions



- ◆ É um módulo que inclui funções de diversos outros blocos: **Assign**, **Delay**, **Size**, **Release**, **Signal**, **Wait**, etc.

## Menu



Menu



- ◆ Permite interação com o usuário para entrada de dados pelo teclado.

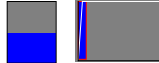
© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Animate



Animate

- ◆ Contém diversos outros módulos de animação entre os quais **Plot**, **Histogram** e **Level**.



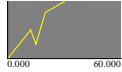
Animate

Number Available

## Container



Container



Container 2

Container

- ◆ Prove a capacidade de modelar processos contínuos tais como tanques e **containers**. Podem ser especificados para atuar como Fonte, Transferência ou Sorvedouro. Consiste de dois símbolos de animação: **Level** e **Plot**.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Expressions

$y=x$

Expressions

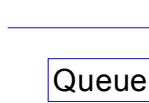
Expressions

- ◆ Define expressões e os valores associados. Expressões são referenciadas no modelo através de seus nomes e podem ser especificadas como vetores ou matrizes.

## Queue



Queue



Queue

Queue 1

- ◆ Define um elemento de fila e cria sua animação. No arena, filas são criadas automaticamente nos módulos que as utilizam e por isso, o módulo **Queue** é raramente necessário.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

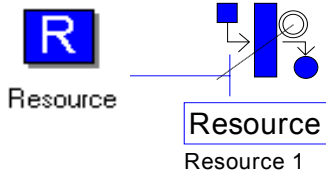
## Recipes



### Recipes

- ◆ São utilizadas para armazenar valores dependentes da localização. A associação de uma **Recipe** a uma estação define valores estáticos a serem utilizados pela estação. Também pode ser associado a módulos **Sequence**.

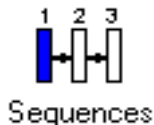
## Resource



- ◆ Define recursos e a animação dos mesmos. Alguns tipos de módulos já possuem recursos definidos internamente, tais como o **Server**.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Sequences



### Sequences

- ◆ É utilizado para definir uma sequência de estações para as entidades fluírem através do modelo. Uma sequência consiste de uma lista ordenada de estações que uma entidade visitará.

## Sets

{ abc }

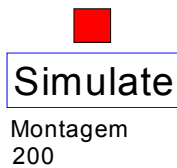
Sets

### Sets

- ◆ Define grupos de elementos similares que podem ser referenciados através de um nome comum e um índice de conjunto. Por exemplo, podemos definir um conjunto de recursos similares e que podem ser alocados indistintamente.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Simulate



- ◆ Especifica o número de replicações da simulação a serem realizadas, podendo podendo-se definir, entre outros parâmetros, o tempo máximo de duração da simulação e condição de término da simulação.

## Statics

**X=4**

Statics

Statics

- ◆ Valores podem ser armazenados em atributos, variáveis, expressões ou estático. Um valor estático pode mudar baseado na localização da entidade que está referenciando o estático.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Statistics



Statistics

- ◆ É utilizado para definir estatísticas que serão coletadas durante a simulação e para especificar os arquivos de saída onde estas observações estatísticas serão escritas para uso do **Arena Output Analyzer**.

## Storage



- ◆ O módulo **Storage** define um local de armazenamento e gera a animação deste local. O **Arena** cria automaticamente lugares de armazenamento sempre que um modulo o exige, o que tornam raras as situações em este módulo é necessário.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Variables



- ◆ Especifica variáveis globais definidas pelo usuário, bem como seus valores iniciais. Podem ser especificadas como vetores ou matrizes.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Alguns Módulos do Painel Support

### Chance



■ Chance

- ◆ Provê um meio de se definir diferentes direções de fluxo para entidades definindo-se probabilidades para seguir cada uma das direções.

### Choose



■ Choose

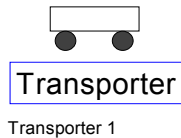
- ◆ Provê um meio de se definir diferentes direções de fluxo para entidades baseado em condições lógicas da forma verificadas através de **if else**.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Alguns Módulos do Painel Transport

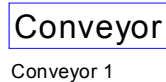
### Transporter

- ◆ Permite a definição de um dispositivo de transporte para movimentação de entidades de uma estação a outra.











### Conveyor

- ◆ Permite a definição de uma esteira de transporte de entidades para movimentação entre as estações.



© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados









## Animações

	Queue		Station
	Storage		Intersection
	Parking		Route
	Seize (reserva recurso)		Segment

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados





## Animações

	Distance		Date
	Network		Level
	Variable		Histogram
	Clock		Plot

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## Animações

	Entity		Promote Path
	Transporter		
	Resource		
	Global		

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

## ***Referências***

- ◆ Kelton, W. D., Sadowski, R. P., Sadowski, D. A.,  
"Simulation with ARENA", WCB McGraw-Hill, ISBN 0-07-027509-2, 1998. Inclui CD com o ARENA versão 3.01 Acadêmica.
- ◆ Prado, D. "Usando o ARENA em Simulação", Editora de Desenvolvimento Gerencial, ISBN 85-86948-19-5, 1999. Inclui CD com o ARENA versão 3.2 de demonstração.

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados

***Fim do Módulo  
ARENA***

© Copyright Graça Bressan/Larc 2001 - Todos os direitos são reservados